Technical Report TREE-0106

Nonparametric Estimation of Internal Delay Densities from Unicast End-to-end Measurement

M. Coates, R. Nowak, Y. Tsang*

November 19, 2001

Abstract

The substantial overhead of performing internal monitoring motivates techniques for inferring spatially localized information about network performance using only end-to-end measurements. In this paper, we present a novel methodology for inferring queueing delay distributions across internal links in the network based solely on unicast, end-to-end measurements. A key feature of our new approach is that it is nonparametric, meaning that no *a priori* limit is placed on the number of unknown parameters used to model the delay distributions. We advocate this nonparametric method because it is our experience that no sufficiently simple parametric model is capable of portraying the wide variety of internal delay distributions. The methodology is formulated according to a recently proposed nonparametric, wavelet-based density estimation method in combination with an expectation-maximization optimization algorithm that employs a new fast Fourier transform implementation. We perform `ns` simulations to verify the accuracy of the estimation procedure.

1 Introduction

Spatially localized information about network performance plays an important role in isolation of network congestion and detection of performance degradation. Routing algorithms, servicing strategies, security programs and performance verification can benefit from monitoring techniques that report such information. Monitoring can be performed internally, but it is impractical to directly measure traffic characteristics at all internal devices for a number of reasons [1]. This has prompted several groups to investigate methods for inferring internal network behavior based on "external" end-to-end network measurements [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. This problem is often referred to as *network tomography*.

*The authors are with the Department of Electrical and Computer Engineering, Rice University, Houston, Texas, USA. E-mail: {mcoates,nowak,ytsang}@rice.edu .

Queuing delays are one of the most critical performance characteristics. Optimizing communication network routing and service strategies requires knowledge of the queueing delay at different points in the network. Measuring end-to-end (source to receivers) delays using timestamps [8, 11, 12] is relatively easy and inexpensive in comparison to internal measurement, although there are of course measurement issues that must be addressed. It is natural to consider the following problem: from end-to-end measurements can we resolve the queueing delay experienced at internal points in the network? More precisely, the goal of the network tomography problem considered in this paper is to estimate the probability distribution of the queueing delay on each link based on end-to-end packet pair measurements.

1.1 Contribution

In this paper, we describe a *nonparametric* framework for the inference of internal delay distributions based on unicast end-to-end measurement. By nonparametric we mean that no *a priori* limit is placed on the number of parameters or degrees of freedom used to describe the observed delay measurements. Most work to date in network tomography is based on *parametric* models. Parametric models assume that the measured traffic data depends on a finite number of parameters. For example, earlier work in delay distribution estimation has been based on discretized (or quantized) delay measurements, with internal delay distributions modelled as discrete probability mass functions (pmfs) [1, 4, 5]. In this context, the parameters are simply the probabilities associated with each pmf. It has been our experience, however, that no sufficiently simple parametric model is capable of portraying the wide variety of internal delay distributions observed in practice, thus motivating the consideration of nonparametric or continuous models.

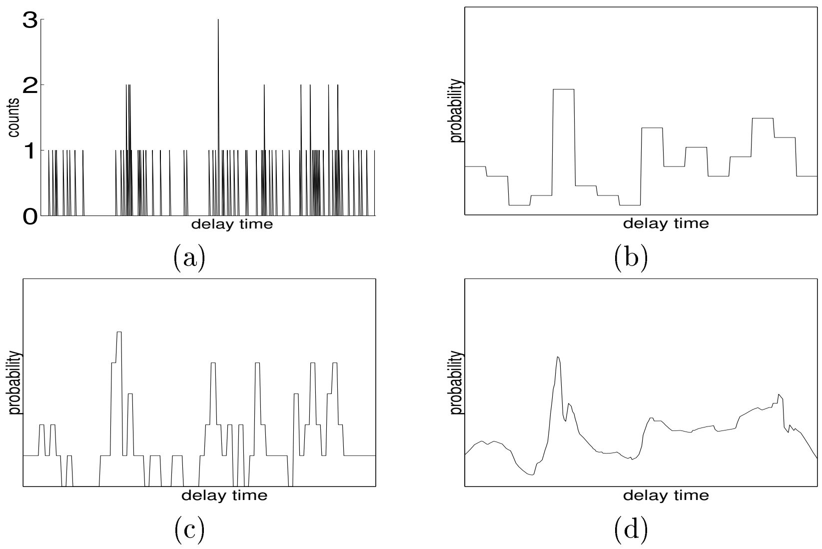


Figure 1: (a) ns delay measurements on link 9 for network depicted in Fig. 2. Horizontal axis is (discretized) delay time and vertical axis denotes the number of occurences of a particular delay measurement. (b) Discretized pmf with 16 equal-width bins. (c) Discretized pmf with 64 bins. (d) Nonparametric density estimate proposed in this paper.

The selection of the "right" quantization level (or bin size) presents a complex trade-off between computational complexity and estimation accuracy. Moreover, finer quantization does not necessarily imply that more accurate estimates will be obtained. Finer quantization provides more flexibility in the estimator (which in turn leads to a reduction in bias) but at the same time has the effect of increasing the number of parameters and hence the variance of the estimator (since all inferences are based on a finite number of measurements in practice). This is the classic bias-variance trade-off in parametric statistics. This effect is clearly evident in the `ns` results depicted in Fig. 1. Note that the 16 bin pmf misses some of the detail and structure in the distribution of delay measurements, whereas the 64 bin pmf appears to overfit the data. Indeed, if we select no binning at all, beyond the initial binning of delay measurements in Fig. 1(a), then the MLE pmf is simply a normalized (by the total number of measurements) version of the data histogram in Fig. 1(a). A procedure for automatically selecting the best bin size for each link was proposed in [1]. This is important since links may differ dramatically in terms of speed and buffer sizes. However, from Fig. 1 it is clear that there may be no one bin size that is adequate for a given link; a finer bin size might be required to capture the detailed structure in "peaky" regions of the density and a coarser bin might be preferable in smoother regions to reduce the variance of the estimate.

Fully continuous or nonparametric methods are preferable in networking applications because in principle these methods place little or no prior assumptions on the form of the delay distributions. Continuous methods have been proposed based on inverse Laplace transforms [1] and characteristic functions [13], but these methods are computationally quite prohibitive and furthermore they do not solve the bias-variance trade-off. In fact, without some assumptions of regularity or "smoothness" of the underlying continuous density, continuous density estimation methods are severely ill-posed and generally lead to poor estimates [14].

We adopt a different approach in this paper drawing on recent developments from the field of nonparametric statistics. We apply the multiscale maximum penalized likelihood estimation (MMPLE) procedure developed in [15] in combination with a tomographic inference strategy known as the Expectation-Maximization (EM) algorithm [4]. Unlike the previous parametric approach in [4], which suffers from the pitfalls associated with all parametric methods, the new methodology is fully nonparameteric and requires no tuning or ad hoc adjustment. Fig. 1(d) illustrates the performance of the new delay estimator. The nonparametric estimator is based on wavelet denoising principles and automatically balances the trade-off between fidelity to the data and complexity of the estimated density. The new methodology offers several significant advantages over existing methods: (1) the estimation procedure is nonparametric and very flexible, in that it is capable of recovering densities from a broad range of function spaces including Bound Variation (BV) functions and Besov spaces, which include both smooth and piecewise smooth densities; (2) the use of a MMPLE scheme provides a computationally fast method for balancing the bias-variance trade-off and has been shown to be nearly optimal for density estimation in the abovementioned function spaces [15]; (3) we develop a new fast Fourier transform based implementation of the EM algorithm for the network tomography problem which, in combination with MMPLE, leads to a worst-case overall complexity of $O(MN^2 \log N)$ where M is the number of links in the network and N is the number of delay measurements. In general, the complexity is substantially less than this (see Section 3.4 for clarification). We demonstrate the flexibility and accuracy of the nonparametric approach through `ns` simulation.

1.2 Related Work

Lo Presti *et al.* have outlined a framework for the inference of internal queuing delay distributions based on multicast end-to-end measurement [1]. Procedures for estimating low order moments such as link delay variances have also been developed [16]. The multicast framework has the advantage of scalability (each measurement probe provides some information about all links in the considered network) and guaranteed, structured correlation between the delay measurements at different receivers. However, multicast is not supported by all networks and there is evidence that routers treat multicast packets differently from the unicast packets that make up the majority of network traffic [6]. These concerns motivate the development of an inference framework based on unicast measurement. However, an important new consideration arises in the unicast setting. That is, for a fixed measurement overhead, multicast measurement provides much more data than unicast. This means that if the framework of [1] were adapted to unicast measurement, as suggested in [6], it would need to perform with significantly less information available.

Lai and Baker [8] have implemented `nettimer`, a procedure that estimates link-level bandwidth. Similar in nature to `pathchar` [17], it exploits the time-to-live field of packets to collect informative measurements. `nettimer` generates accurate estimates of bandwidths (particularly when they are small), although requires a relatively large number of measurement packets. Theoretically, it could be used to estimate queuing delays, but to our knowledge there has been no experimental work exploring its performance. The number of measurement packets needed for estimation may prove prohibitive given the short duration over which delay distributions are generally stable. It would seem that network utilization would need to be low in order to achieve reliable estimates.

Shih and Hero have developed a method for estimation of the link delay cumulant generating functions (CGFs) [13, 18]. The CGFs have the advantage of being additive over a path of several links in contrast to the convolutional way in which link delay pmfs combine to form end-to-end delay pmfs. Based on the disentangled CGFs, it is straightforward to reconstruct the delay distributions. This technique has the benefit of imposing no discretization, but does not impose smoothness constraints, leading to an ill-posed problem when data is limited. The chief disadvantage of the technique is that in order for all links to be resolved, internal measurements must be available or a tool such as `nettimer` must be used.

Coates and Nowak have described a sequential Monte Carlo-based internal delay estimation framework in [4, 5]. This framework directly addresses the time-varying nature of network delay behaviour. In this approach, a fine-level of quantization can be imposed, and smoothness is incorporated through the adoption of a slowly-varying time-dependent Bayesian prior. However, the parameters associated with the prior introduce a potentially undesirable parametric nature to the estimation task.

1.3 Paper Structure

The remainder of the paper is structured in the following manner. In Section 2 we describe the measurement framework, modelling assumptions and implementation requirements. In Section 3 we describe the inference methodology, detailing the MMPLE procedure and EM algorithm. In Section 4 we describe the results of `ns` experiments designed to explore the performance of the methodology. In Section 5, we make some concluding remarks and indicate avenues of future research.

2 Measurement Framework

Throughout this paper, we concentrate on networks comprised of a single source transmitting measurement probes to multiple receivers. There is no difficulty extending the approach to measurements made at multiple sources, although care must be taken that measurements are sufficiently separated for independence assumptions to hold. We assume that the topology is fixed throughout the measurement period, but straightforward extensions can account for changes in topology over coarse time scales.

For the networks we consider, standard network routing protocols produce a tree-structured topology, with the *source* at the root and the *receivers* at the leaves. A network with six receivers is depicted in Fig. 2. The nodes between the source and receivers represent internal *routers*. Connections between the source, routers, and receivers are called *links*. Each link between routers may be a direct connection, or there may be "hidden" routers (where no branching occurs) along the link that are not explicit in our representation. We adopt the notation that link i connects node i (below) to its parent node $\omega(i)$.

The basic measurement and inference idea is quite straightforward. Suppose two closely time-spaced (back-to-back) packets are sent from the source to two different receivers. The paths to these receivers traverse a common set of links, but at some point the two paths diverge (as the tree branches). The two packets should experience approximately the same delay on each shared link in their path. This facilitates the resolution of the delays on each link.

We distinguish between a *measurement* period and an *inference period*. The measurement period is the time period over which all measurements are collected. The inference period is some time window within the measurement period; the window duration is dictated by the degree of network stationarity and only the measurements collected in this window are used to perform inference. In order to achieve estimates over the entire measurement period, multiple inferences must be performed using different, potentially overlapping inference windows.

We collect measurements of the end-to-end delays from source to receivers, and we index the packet pair measurements by $k = 1, \dots, N$. For the k-th packet pair measurement, let $y_1(k)$ and

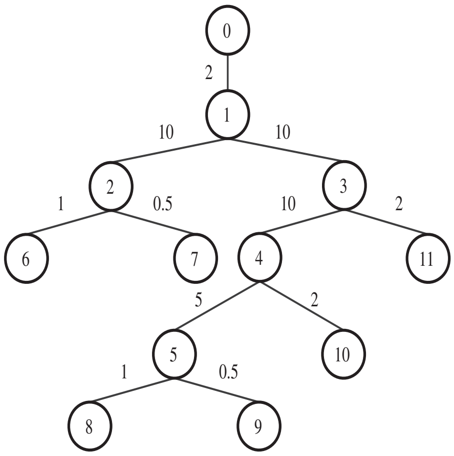


Figure 2: Tree-structured network topology used for ns simulation experiments. Source (node 0) transmits to 6 receivers (nodes 6-11). Link speeds in Mb/s are shown next to the links. Link i connects node i to its parent node $\omega(i)$, e.g. link 9 connects nodes 5 and 9.

$y_2(k)$ denote the two end-to-end delays measured. The ordering 1 and 2 is *arbitrary*; the indices are randomly selected with no dependence the order in which the packets were sent from the source. This will be important in dealing with discrepencies between the delays experienced by the two packets on shared links, which will be discussed in greater detail in Section 2.1. In this paper, we do not consider the case in which one or both of the packets is dropped (lost). We simply discard packet pairs in which a loss occurs. However, it is possible to extend our approach to include losses. Since we are interested in inferring queuing delay, our first step is to extract what we perceive as the minimum delay (propagation + transmission) on each measurement path. This is estimated as the smallest delay measurement we acquire on the path during the measurement period. We assume that the true minimum delay is observed over the measurement period. If this is not the case, then queuing delay is systematically underestimated for links on the affected path.

Our goal is a nonparametric estimate of the delay distributions on each link. Clearly it is impossible to completely determine an infinite dimensional density function from a finite number of delay measurements, but we require that as the number of delay measurements increases so does the accuracy of our estimation procedure. Thus, we adopt the following procedure. The end-to-end delay measurements are binned, *but* the number of bins is chosen to be equal to or greater than the number of delay measurements. We stress that this is not a parametric step. This means that there is less than one measurement per bin, on average, and hence we do not lump or group delays in a potentially artificial, prescribed fashion. Thus, we place no prior restriction on the form of the density estimator; the more measurements one has, the more one can resolve the structural nuances of the delay densities.

In practice we choose the number of bins to be the smallest power of two greater than or equal to the number of measurements (facilitating certain processing steps to be described later). We upper bound the maximum delay on any one link by the maximum end-to-end delay along the path(s) that include that link. Let $d_{\max}$ denote this upper bound for a particular link and let K be the smallest power of 2 that is greater than or equal to the number of measurement packets N. The bin width for the link is then set at $d_{\max}/K$. This procedure is conservative, in that the estimated $d_{\max}$ may be substantially larger than the true maximum queuing delay. It may be preferable to use previous link-delay estimates or bandwidth estimates from a procedure such as `nettimer` [8] to gauge the maximum delay on any link.

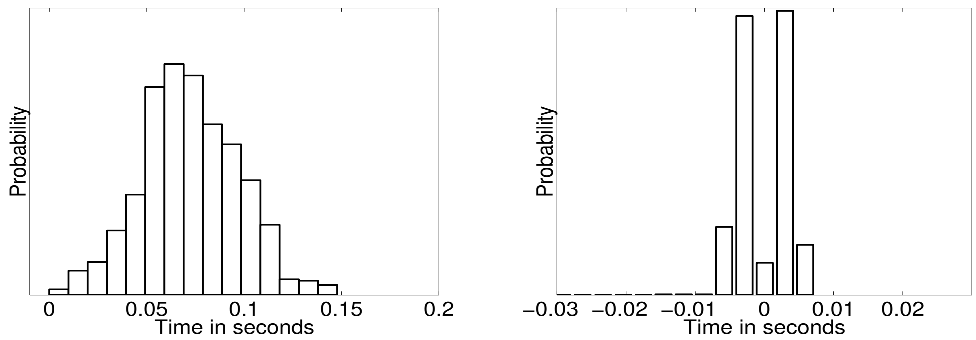
At this stage, each end-to-end measurement has been ascribed a discrete number between 0 and $L \times (K-1)$, where L is the maximum path length in the network. To illustrate our inference methodology in its simplest form, suppose that we send many packet pairs to receivers 6 and 7 in Fig. 2 and measure the delays experienced by each packet. Each measurement consists of a pair of delays, one being the delay to receiver 6 and the other the delay to receiver 7. From these measurements, collect events where '0' delay (a delay in bin zero) is measured at receiver 6. Now, assuming that the delay is the same for both packets on the common links (1 and 2 in this case), any "additional" delay observed to the receiver at 7 can be attributed to link 7 alone. We can then build a histogram estimate of the delay pmf for link 7. This simple idea can be extended to obtain estimators for the delay distributions on all links which also make use of *all* the measured data (not just special cases like the one above). In Section 3 , we describe the large-scale inference procedure in detail.

Suppose the network is stationary over each inference period, the delays are identical on shared links, and the true delay pmfs are strictly positive and canonical (there is some mass in the zero delay bin). In addition, suppose that the link delays experienced by an individual packet are independent of one another. Then, based on the multicast analysis made in [1], one can show that the true distributions can be uniquely identified from such end-to-end measurements (as the number of measurements tends to infinity).

2.1 Model Assumptions

There are several assumptions in the framework that are worthy of discussion. Firstly, we assume spatial independence of delay. Delay on neighbouring links is generally correlated to a greater or lesser extent depending on the amount of shared traffic. In the `ns` experiments discussed in Section 4, correlation of delays is observed. In the presence of weak correlation, our framework is able to derive good estimates of the delay distributions. As the correlation grows stronger, we see a gradual increase of bias in the estimates. We also assume temporal independence (successive probes across the same link experience indepent delays). Temporal dependence was observed in [1] and in our experiments; indeed it is exploited in [5]. As in [1], the maximum likelihood estimator we employ remains consistent in the presence of temporal dependence, but the convergence rate slows. It does not have a dramatic effect on the performance of the estimator.

Finally, our framework hinges on an assumption that packets in a pair experience a common delay on shared links. If the delays are identical on shared links, then the difference between the two delay measurements can be attributed solely to the delays experienced on unshared links in the two paths. This is the key to resolving the delays on a link by link basis. However, in practice the two packets may experience slightly different delays on shared links due to the fact that one packet precedes the other in the queues and additional packets may intervene between the two. The nature of this delay differential is exposed in Fig. 3, which shows the histogram of the difference between the end-to-end delays of two closely-spaced packets sent to the same receiver over the Internet. This histogram is constructed from back-to-back packet pair measurements using the `netdyn` tool [12]. Ideally, the delays should be identical, but we see a small discrepancy between the two. The second packet in the pair typically experiences a slightly greater delay. However, recall that the ordering of the packets was arbitrary in our recording process. In effect then, the discrepancies between the delays on shared links adds an approximately zero mean error to the difference between the two end-to-end measurements. We clearly see the symmetric zero-mean nature in the empirical data shown in Fig. 3, and we have observed similar behavior in all our measurements and simulations. This "noise" produces a smoothing (or blurring) in the inferred delay pmfs. Nonetheless, because the errors are roughly zero mean, we can still use the estimated delay pmfs to obtain approximately unbiased estimates of the expected delay on each link or the locations of modes in the density, for example.



(a) (b)

Figure 3: (a) End-to-end delay histogram (packets sent from Rice University to Michigan State University). (b) Difference between delays of the two packets in packet pairs. Measurements were made using the *netdyn* tool.

2.2 Measurement Requirements

The delay inference framework requires knowledge of the (logical) topology of the network and the capability to perform one-way delay measurements. We perform the construction of the topology using a modified, lightweight version of `traceroute` [19, 20]. Collection of one-way delay measurements requires that the receivers cooperate with the source.

We do not necessarily require that clocks at the source and receivers be synchronised, but we do require that the disparity between clocks remain very nearly constant over the measurement period. In this way we can be sure that subtracting the estimated minimum delay does not induce bias in our estimates. A further difficulty lies in clock resolution. Clocks must be able to resolve delay sufficiently accurately that the potential error does not overwhelm the true delay value. Deployment of Global Positioning System (GPS) devices allows these clock difficulties to be avoided, as it provides synchronised measurements to within tenths of microseconds. Alternatively, delay measurements can be adjusted using algorithms developed to detect and compensate for clock adjustments and rate discrepancies [21, 22]. In this paper, we assume that synchronised measurements are available.

3 Delay Distribution Inference

We commence the description of our inference framework by formalising our measurement and modelling notation. Let $p_i = \{p_{i,0} \dots, p_{i,K-1}\}$ denote the probabilities of a delay of $0, 1, \dots, K-1$ time units, respectively, on link i. We denote the packet pair measurements $\boldsymbol{y} \equiv \{y_1(k), y_2(k)\}_{k=1}^{N}$.

In general, only a relatively small amount of data can be collected over the period when delay distributions can be assumed approximately stationary. A natural estimate would be the maximum likelihood estimates (MLEs) of $\boldsymbol{p} \equiv \{p_i\}$, the collection of all delay pmfs. However, when we strive for nonparametric estimation by using a number of bins that is equal to or larger than the number of measurements, the problem is ill-posed and the MLE tends to overfit to the probe data (see Fig. 1(a)), producing highly variable estimates that do not accurately reflect the delay distribution of the traffic at large. High variance manifests itself in irregular, noisy-looking estimates [14]. One way to reduce this irregularity is to maximize a penalized likelihood (see Fig. 1(d)). We replace the maximum (log) likelihood objective function $L(\boldsymbol{p}) = \log l(\boldsymbol{y}|\boldsymbol{p})$ with an objective function of the form:

$$L(\boldsymbol{p}) - \mathrm{pen}(\boldsymbol{p}) \tag{1}$$

where $\mathrm{pen}(\boldsymbol{p})$ is a non-negative real-valued functional that penalizes the "roughness" (*complexity*) of $\boldsymbol{p}$. A small value of $\mathrm{pen}(\boldsymbol{p})$ indicates that $\boldsymbol{p}$ is a smooth (simple) estimate; a large value indicates that $\boldsymbol{p}$ is rough (complicated). The maximization of this penalized log-likelihood involves a trade-off between fidelity to the data (large $L(\boldsymbol{p})$) and smoothness or simplicity (small $\mathrm{pen}(\boldsymbol{p})$). We will describe a specific choice of penalty functional in Section 3.2. Before moving to that, however, we will quickly formulate the basic likelihood fucntion and motivate the adoption of an EM algorithm for optimization.

3.1 Likelihood Function

Under the assumption of spatial independence, the likelihood of each delay measurement $\{y_1(k), y_2(k)\}$ is parameterized by a convolution of the pmfs in the path from the source to receiver. With our modelling constraint that packets in a pair experience the same delay on shared links, the likelihood of the two measurements made by the k-th packet pair is:

$$l(y_1(k), y_2(k)|\boldsymbol{p}) \quad =$$

$$\sum_j \rho_{c,k}(j)\rho_{1,k}\left(y_1(k)-j\right)\rho_{2,k}\left(y_2(k)-j\right). \tag{2}$$

The range of the summation is determined by the ranges of the pmfs $\rho_{c,k}$, $\rho_{1,k}$ and $\rho_{2,k}$. The pmf $\rho_{c,k}$ is the convolution of the pmfs of the links on the shared path of the two packets, e.g. $\rho_{c,k} = p_1 * p_2$ for a 6-7 pair in Fig. 2 with $*$ denoting convolution). The pmf $\rho_{1,k}$ (resp. $\rho_{2,k}$) is the convolution of the pmfs on the links traversed only by the packet that measures $y_1(k)$ (resp. $y_2(k)$). The joint likelihood $l(\boldsymbol{y}|\boldsymbol{p})$ of all measurements is equal to a product of the individual likelihoods:

$$l(\boldsymbol{y}|\boldsymbol{p}) = \prod_{k=1}^{N} l(y_1(k), y_2(k)|\boldsymbol{p}). \tag{3}$$

The presence of convolved link pmfs in the likelihood of each measurement (2) results in an objective function that cannot be maximized analytically. The maximization of the likelihood function requires numerical optimization, and an EM algorithm [23] is an attractive strategy for this purpose. Before giving the details of the algorithm, we briefly review the MMPLE nonparametric density estimation procedure employed in our framework.

3.2 MMPLE Density Estimation

Here we briefly outline the MMPLE density estimation procedure developed in [15]. To introduce the idea, we consider a case where the link delays have been directly measured (we will handle the tomographic case using the EM algorithm outlined in the next section). Let $z_i(k)$, $k = 1, \ldots, N_i$, denote a set of delay measurements for a particular link i. We assume that these measurements are independent and identically distributed according to a continuous delay density $p(t)$, where without loss of generality we assume that $t \in [0,1]$ (for convenience of exposition we take the maximum delay to be unity). Define a discrete pmf via $p_{i,j} = \int_{(j)/K}^{(j+1)/K} p(t)dt$, $j = 0, \ldots, K-1$, where K is the smallest power of two greater than or equal to N_i. It follows that the number of measurements falling in the interval $[\frac{j}{K}, \frac{(j+1)}{K}]$, denoted $m_{i,j}$, is multinomially distributed [24], i.e., $\{m_{i,j}\} \sim \text{Multinomial}(N_i; \{p_{i,j}\})$. The MMPLE estimator maximizes the following criterion with respect to $\{p_{i,j}\}$:

$$\log \text{Multinomial}(N_i; \{p_{i,j}\}) - \text{pen}(\{p_{i,j}\}), \tag{4}$$

where

$$\text{pen}(\{p_{i,j}\}) \equiv \frac{1}{2}\log(N) \times \#_i, \tag{5}$$

where $\#_i$ is the number of non-zero coefficients in the discrete Haar wavelet transform of the pmf $\{p_{i,j}\}$ (see [15] for details). This number reflects the irregularity and complexity of the pmf — the larger the value of $\#_i$, the more "bumps" in the pmf. There are two important features of the MMPLE: (1) the global maximizer can be computed in $O(K)$ operations; (2) the MMPLE is nearly minimax optimal in the rate of convergence over a broad class of function spaces. The near minimax optimality implies that the rate at which the estimator converges to the true continuous density (as a function of the number of measurements N) cannot be significantly improved upon. More complicated and computationally intensive procedures will not significantly outperform the MMPLE. The optimization is carried out by performing a set of K independent generalized likelihood ratio tests. In all results in this paper we employ a *translation-invariant* version of the MMPLE, in which multiple MMPLEs are computed with K different shifted versions of the Haar wavelet basis and the resulting estimates are averaged. This produces a slight improvement over the basic MMPLE and can be efficiently computed in $O(K \log K)$ operations.

3.3 EM Algorithm

The MMPLE methodology can be employed in the tomographic delay estimation case by simply adopting the penalized likelihood criterion:

$$\log l(\boldsymbol{y}|\boldsymbol{p}) - \sum_i \frac{1}{2} \log(N_i) \times \#_i, \tag{6}$$

where N_i denotes the number probe packets passing through link i and $\#_i$ denotes the number of non-zero Haar wavelet coefficients in the delay pmf of link i. The difficulty is that this penalized likelihood function cannot be maximized by a simple set of likelihood ratio tests due to the nonlinear relationship between link delay pmfs and end-to-end measurements $\boldsymbol{y}$. The EM algorithm is an iterative procedure designed to maximize (6) that takes advantage of the $O(K)$ computational simplicity of the MMPLE technique.

The first step in developing an EM algorithm is to propose a suitable *complete data* quantity that simplifies the likelihood function. Let $z_i(k)$ denote the delay on link i for the packets in the k-th pair. Let $\boldsymbol{z}_i = \{z_i(k)\}$ and $\boldsymbol{z} = \{\boldsymbol{z}_i\}$. The link delays $\boldsymbol{z}$ are not observed, and hence $\boldsymbol{z}$ is called the *unobserved data*. Define the *complete data* $\boldsymbol{x} \equiv \{\boldsymbol{y}, \boldsymbol{z}\}$. Note that the complete data likelihood may be factorized as follows:

$$l(\boldsymbol{x}|\boldsymbol{p}) = f(\boldsymbol{y}|\boldsymbol{z})g(\boldsymbol{z}|\boldsymbol{p}),$$

where f is the conditional pmf of $\boldsymbol{y}$ given $\boldsymbol{z}$ (which is a point mass function since $\boldsymbol{z}$ determines $\boldsymbol{y}$), and g is the likelihood of $\boldsymbol{z}$. The factorization shows that $l(\boldsymbol{x}|\boldsymbol{p}) \propto g(\boldsymbol{z}|\boldsymbol{p})$, since $f(\boldsymbol{y}|\boldsymbol{z})$ does not depend on the parameters $\boldsymbol{p}$. Next note that the likelihood

$$g(\boldsymbol{z}|\boldsymbol{p}) = \prod_{i,j} p_{i,j}^{m_{i,j}},$$

where $m_{i,j} \equiv \sum_{k=1}^{N} \mathbf{1}_{z_i(k)=j}$ is the number of packets (out of all the packet pair measurements) that experienced a delay of j on link i; here $\mathbf{1}_A$ denotes the *indicator function* of the event A. Therefore, we have

$$l(\boldsymbol{x}|\boldsymbol{p}) \propto \prod_{i,j} p_{i,j}^{m_{i,j}}.$$

Therefore, if the $m_{i,j}$ were available, then the MLE of $p_{i,j}$ would be simply

$$\widehat{p}_{i,j} = \frac{m_{i,j}}{\sum_{k=0}^{K-1} m_{i,k}}. \tag{7}$$

Similarly, given the $m_{i,j}$ we could directly apply the MMPLE described above (see [15] for implementation details).

The EM algorithm is an iterative method that uses the complete data likelihood function to maximize the log-likelihood function. By suitable modification, it can maximize a penalized log-likelihood objective function instead. Specifically, the modified EM algorithm alternates between computing the conditional expectation of the complete data log likelihood given the observations $\boldsymbol{y}$ and maximizing the sum of this expectation and the imposed complexity penalty ($-\text{pen}(\boldsymbol{p})$) with respect to $\boldsymbol{p}$. Notice that the complete data log likelihood is linear in $\boldsymbol{m}$:

$$\log l(\boldsymbol{x}|\boldsymbol{p}) \propto \sum_{i,j} m_{i,j} \log p_{i,j}\,.$$

Thus, in the E-Step we need only compute the expectation of $\boldsymbol{m} = \{m_{i,j}\}$.

E-Step: Let $\boldsymbol{p}^{(r)}$ denote the value of $\boldsymbol{p}$ after the r-th iteration. Then

$$\begin{aligned}
\widehat{m}_{i,j}^{(r)} &\equiv \boldsymbol{E}_{\boldsymbol{p}^{(r)}}[m_{i,j}|\boldsymbol{y}], \\
&= \boldsymbol{E}_{\boldsymbol{p}^{(r)}}\left[\sum_{k=1}^{N} \mathbf{1}_{\{z_i(k)=j\}}|\boldsymbol{y}\right], \\
&= \sum_{k=1}^{N} \boldsymbol{E}_{\boldsymbol{p}^{(r)}}\left[\mathbf{1}_{\{z_i(k)=j\}}|\boldsymbol{y}\right], \\
&= \sum_{k=1}^{N} \boldsymbol{E}_{\boldsymbol{p}^{(r)}}\left[\mathbf{1}_{\{z_i(k)=j\}}|y_1(k), y_2(k)\right], \\
&= \sum_{k=1}^{N} p^{(r)}\left(z_i(k) = j|y_1(k), y_2(k)\right).
\end{aligned} \tag{8}$$

Thus, the conditional expectation of $\boldsymbol{m}$ can be computed by determining the conditional probabilities above for each packet pair measurement.

M-Step: In the penalized case (6), apply the MMPLE algorithm of [15] to the conditional expectation $\{\widehat{m}_{i,j}^{(r)}\}$. In the case of unpenalized maximum likelihood estimation, simply substitute $\widehat{m}_{i,j}^{(r)}$ in place of $m_{i,j}$ in equation (7).

3.4 Fast Fourier Transform based EM Algorithm

The expectation step of the EM algorithm poses the major portion of the computational burden of the optimization task. It can be performed using a message passing (or upward-downard) procedure [25]. Unfortunately, a straightforward implementation of the message passing procedure, as proposed in [4, 5], has a computational complexity which is $O(MK^3)$, where M is the number of links in the tree and K is the number of bins. This is impractical in our nonparametric setting since K is not fixed, but rather increases in proportion to the number of measurements. In this section, we describe a novel, fast Fourier transform based implementation which is $O(MK^2 \log K)$, a tremendous reduction in complexity when K is large.

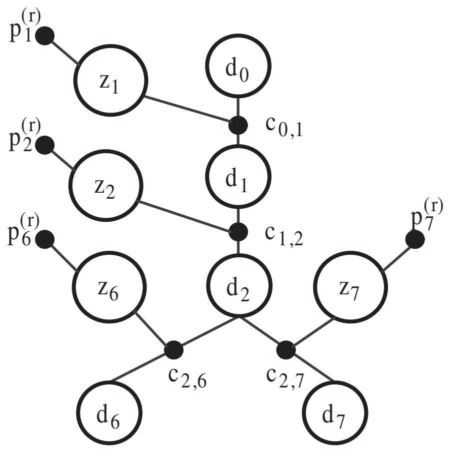


Figure 4: Factor graph used in the message-passing algorithm for a measurement made by a packet pair sent to nodes 6 and 7 in the network of Fig. 2. Measurements are available at nodes 6 and 7; the nodes $p_i^{(r)}$ contain current pmf estimates; and node $c_{a,b}$ indicates the convolutional relationship between nodes d_a, d_b and z_b.

The message passing procedure is based on a factorization of the likelihood function, which can be represented graphically using a *factor graph*. According to (8), our task for each measurement in the r-th iteration of the EM algorithm is to compute $\boldsymbol{p}^{(r)}(z_i = j|y_1, y_2)$ (we have drop the measurement index k for clarity). In the 1980's, Pearl [26] and Spiegelhalter [27] independently developed an exact probability propagation algorithm for inferring the distributions of individual variables in singly-connected graphical models. The basic idea of the algorithm is that each node in the graph propagates its information (a measurement or current pmf estimate in this case) to every other node. Each node then combines all the messages it receives to compute the distribution of its variable.

We illustrate the procedure for a packet pair measurement to nodes 6 and 7 of the network in Fig. 2. For this scenario, the factor graph representation is depicted in Fig. 4. The hollow nodes of the graph represent variables (the link delays $\{z_i\}$ and the cumulative delays $\{d_i\}$ at nodes 0, 1, 2, 6 and 7). The small nodes represent functions — these either indicate functional relationships between variable nodes or carry prior information. In this case, the nodes labelled

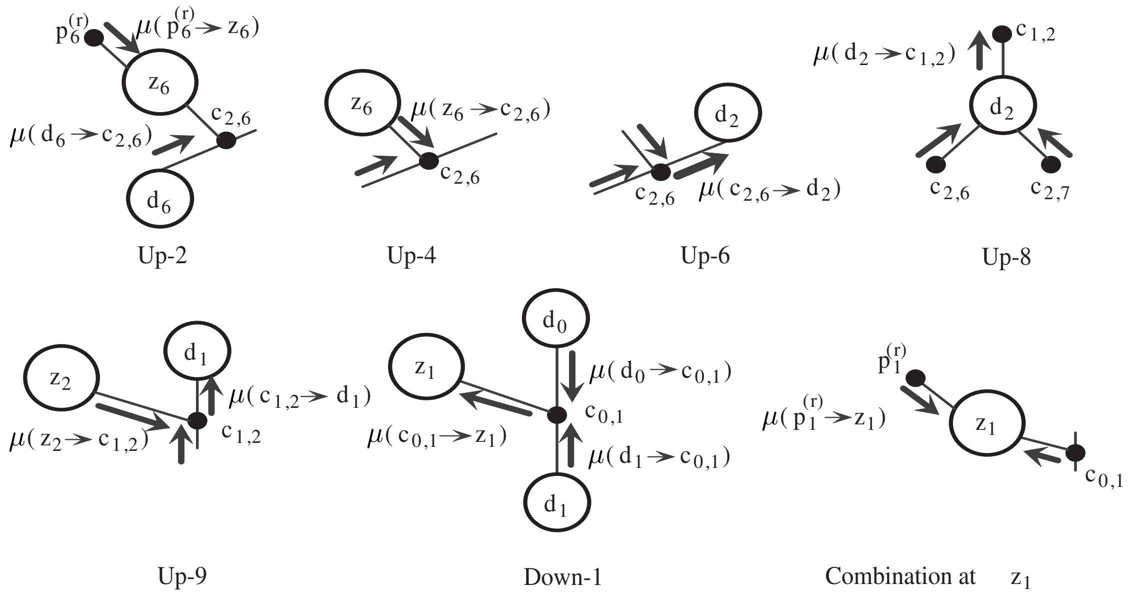


Figure 5: Graphical representation of several important message generation steps in the Upward-Downward message passing algorithm for a measurement made by a packet pair sent to nodes 6 and 7 in the network of Fig. 2. The labels relate to the outline of the algorithm in the text.

$p_i^{(r)}$ carry the current pmf estimates and nodes $c_{i,j}$ represent convolution operators. For example, node $c_{1,2}$ indicates that the accumulated delay pmf at node d_1 is convolved with the link delay pmf z_2 to obtain the accumulated delay pmf at node d_2.

The message passing algorithm can be divided into two sections. In the *upward* stage, all information available at the leaves of the tree is passed up the tree towards the root. In the *downward* stage, the information at the root is passed down to the leaves. In this case, the leaf information includes the measurements and the current pmf estimates. The root information is simply the knowledge that $d_0 = 0$. We use the notation $\mu(a \to b)$ to represent the message that is passed from node a to node b in the graph; each message always takes the form of a pmf. We now provide a brief outline of the messages generated and how they are combined and distributed. Fig. 5 depicts some of the important message generation steps pictorially.

Up-Step:

1. For $i = 1, 2, 6, 7$, the message $\mu(p_i^{(r)} \to z_i)$ is simply the current (r-th iteration) link i pmf estimate.
2. $\mu(d_6 \to c_{2,6})$ is a pmf with mass 1 at bin y_1 (the delay measured at node 6).
3. $\mu(d_7 \to c_{2,7})$ is a pmf with mass 1 at bin y_2 (the delay measured at node 7).
4. $\mu(z_6 \to c_{2,6}) = \mu(p_6^{(r)} \to z_6)$.
5. $\mu(z_7 \to c_{2,7}) = \mu(p_7^{(r)} \to z_7)$.
6. Node $c_{2,6}$ combines its incoming messages according to its convolution function to pass on a message to node d_2; the entry in the j-th bin of the message (a pmf) is $\mu(c_{2,6} \to d_2)(j) = p^{(r)}(z_6 = y_1 - j)$.

7. $\mu(c_{2,7} \to d_2)(j) = p^{(r)}(z_7 = y_2 - j)$.
8. Node d_2 combines the incoming messages by multiplying them together. $\mu(d_2 \to c_{1,2})(j) = \mu(c_{2,6} \to d_2)(j) \times \mu(c_{2,7} \to d_2)(j)$.
9. According to the convolution function at $c_{1,2}$, the message $\mu(c_{1,2} \to d_1)(j) = \sum_{k \geq j} p^{(r)}(z_2 = k - j)\mu(c_{1,2} \to d_2)(k)$.
10. $\mu(d_1 \to c_{0,1})(j) = \mu(c_{1,2} \to d_1)(j)$.

Down-Step: 1. The message $\mu(d_0 \to c_{0,1})$ is a pmf with mass only at 0.

2. The message from $c_{0,1}$ to z_1 combines the upward and downward messages entering $c_{0,1}$ according to the convolutional rule. $\mu(c_{0,1} \to z_1)(j) = \sum_{k \leq j} \mu(d_0 \to c_{0,1})(k) \times \mu(d_2 \to c_{0,1})(j - k)$.
3. Continue in a similar fashion down the tree.

At each variable node z_i, the incoming messages are multiplied together to calculate the local distributions for the delay pmf variables. For example, multiplying the two messages flowing into node z_1 together gives $p^{(r)}\,(z_1 = j|y_1, y_2) = \mu(c_{0,1} \to z_1)(j) \times \mu(p_1^{(r)} \to (z_1))(j)$.

It is the summations in steps Up-9 and Down-2 that introduce the complexity of $O(K^2)$. We avoid this $O(K^2)$ computation by noting that messages involving such summation can be written as a convolutions *provided* we first time-reverse one of the pmfs involved. By time-reversal, we mean that $\widetilde{p}^{(r)}(z_i = j) = p^{(r)}(z_i = K - 1 - j)$ where $\tilde{}$ denotes time-reversal. For example, we can write:

$$\mu(c_{1,2} \to d_1)(j) = [\mu(d_2 \to c_{1,2}) * \tilde{p}\,(r)(z_2)]\,(K - 1 + j). \tag{9}$$

By taking Fourier transforms, the convolution can be implemented as a product in the Fourier domain, reducing the computational complexity to $O(MK \log K)$ per observation.

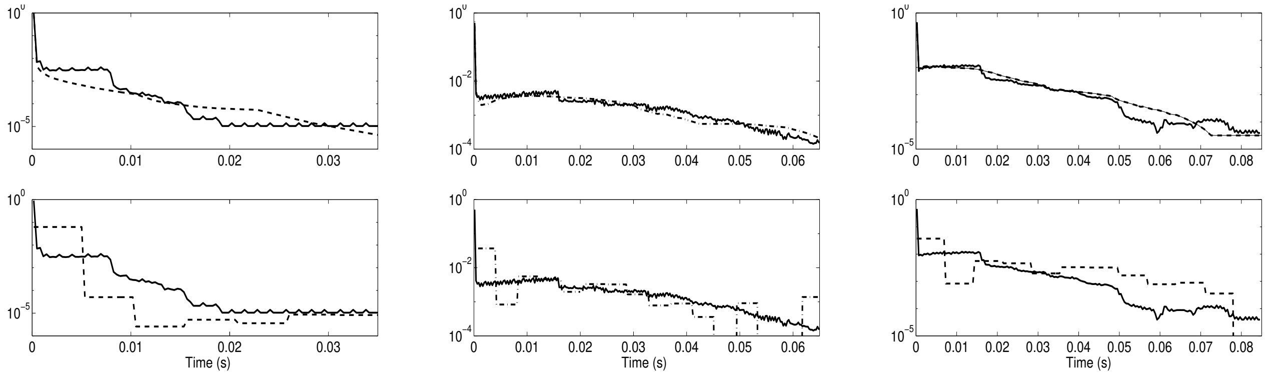
The K^2 factor in the complexity dominates the other terms. However, further computational savings can be introduced by exploiting the additive nature of the Fourier transform. This has the effect of replacing the K^2 factor by $(K/M_{int})^2$, where M_{int} is the number of internal nodes in the network. This can be a substantial; for example in the network of Fig. 2 with 1000 packet-pairs, the necessary compution is reduced from $c \times 1000000$ operations to $c \times 40000$, where c is some constant.

4 Simulation Experiments

In order to verify the performance of our estimation methodology, we conducted `ns` simulation experiments using the network depicted in Fig. 2. Interior links in the network have higher capacity (5-10 Mb/sec) and propagation delay (50 ms) than the edge links (0.5-2 Mb/sec and 10 ms). Queues are FIFO with space for 35 packets. Node 0 generates a 12.8 Kbit/s probing stream comprised of UDP packet-pair probes (40 bytes each). Packet-pair sending times are generated according to a Poisson process; the mean time-spacing is 50 ms. The probe-stream requires less than 1% of any link's capacity. Background traffic comprises a mixture of infinite data-source TCP (FTP) connections, exponential on-off sources using UDP, and multiple short-duration TCP connections. Averaged over the simulations, link utilisation ranged between 10 and 60 percent, and loss rates ranged from 0 to 2 percent.

The network was simulated for multiple two minute measurement periods; from within each measurement period an inference period of 25 seconds was isolated for analysis. This time duration corresponds to 500 packet-pairs (assuming no probes are lost). Throughout the inference period, queue lengths in the network were determined at a fine time scale by monitoring the arrivals of every packet at each queue. A "true" pmf for each link was formed by calculating delays from queue lengths and link capacities, quantizing and forming a histogram. When generating this true pmf, so much data is available that the quantization can be very fine (constructing an excellent estimate of the delay density) without affecting estimation stability.

In Fig. 6, we show the results of one experiment, comparing the true pmfs to the nonparametric MMPLE estimator and the MLE estimator of [4] (the estimation technique in [1] also returns the MLE, although it is devised in a different setting). We display results for the lower bandwidth links because for our experimental set-up, queueing delay was concentrated in these links. There is substantial mass in the tails of thes pmfs and we can evaluate how well the pmf estimates generated by our proposed methodology estimates match the tails. In the higher bandwidth links, there is much less mass in the pmf tails. For these links, both the MLE and MMPLE estimates match the true pmf where probability mass is concentrated, but there is insufficient information to closely match the tails. We calculated the MLE for a variety of bin sizes, but show the bin size that achieved the best fit to the true pmf (in this case 16 bins). The nonparametric estimator was calculated from 512 bins.



(a) (b) (c)

Figure 6: Comparison between true pmfs (solid) and estimated pmfs (dashed). Top panel shows true pmf and MMPLE (calculated using 512 bins); bottom panel shows true pmf and MLE (calculated using 16 bins). 16 bins is determined as the bin size at which the MLE obtains the best fit. (a) Link 5. (b) Link 7. (c) Link 9.

In Fig. 7, we plot the magnitude of the L_1 error norm between the true pmf and the MMPLE

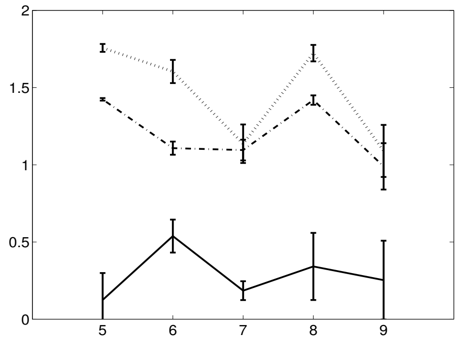


Figure 7: L_1 error criterion averaged over 25 simulations (means and standard deviation). Solid line is MMPLE, dashed line is MLE (16 bins), dotted line MLE (64 bins).

for the links in the network, as averaged over 25 simulations. Also shown are the results for the MLE for medium (64 bins) and large (16 bins) bin sizes. The L_1 error norm is simply the sum of the absolute difference between the estimated pmf and the true pmf over the K bins (MLE estimates made with fewer bins are appropriately converted). As discussed in [24], the L_1 error criterion is a common measure of the performance of a density estimate. The advantage of such as a measure as opposed to a mean-squared error criterion is that more attention is paid to the tails of the distributions. It also enjoys several theoretical advantages over other measures [24].

As is evident from the two figures, the MMPLE technique generates estimates which are smooth, close fits to the true pmfs. In order to introduce some degree of smoothness, MLE estimates must be calculated using a large bin size, resulting in an inability to capture the finer details of a pmf.

When the amount of probing that can be performed is limited, we believe that the most substantial source of error is the intrinsic variability in probe measurements. Another potential source of error is the discrepancy between the delays experienced by the two packets in each pair on their common path. We therefore examined the extent and effect of the delay discrepancy; with 512 bins, the overwhelming majority of the discrepancy was concentrated in 0-3 bins, with a maximum value of 16 bins. The effect of these discrepancies on the quality of the estimates is relatively minor when such a small amount of data is available for inference. If we use our queue monitoring to construct an artificial set of measurements (thereby providing ideal packet-pair probe measurements to our algorithms), the estimates we obtain are very similar to those we report here.

5 Conclusions

We have described a *nonparametric* framework for the inference of internal delay distributions based on unicast end-to-end measurement. The key features of the framework are its flexibility (the ability to capture fine details and smooth regions) and the introduction of a complexity penalization that allows smooth, accurate estimates to be generated even when the amount of data is very small. The basic MMPLE framework developed here could be extended to the multicast approach suggested in [6] and may also be applicable in time-varying contexts like those considered in [4, 5].

References

[1] F. Lo Presti, N.G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," Tech. Rep., University of Massachusetts, 1999.

[2] R. Cáceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Trans. Info. Theory*, vol. 45, no. 7, pp. 2462–2480, November 1999.

[3] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," in *ITC Seminar on IP Traffic, Measurement and Modelling*, Monterey, CA, Sep. 2000.

[4] M. Coates and R. Nowak, "Network delay distribution inference from end-to-end unicast measurement," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, May 2001.

[5] M. Coates and R. Nowak, "Sequential Monte Carlo inference of internal delays in nonstationary communication networks," to appear in *IEEE Trans. Signal Processing, Special Issue on Monte Carlo Methods for Statistical Signal Processing*, 2001.

[6] N.G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, April 2001.

[7] A. Bestavros K. Harfoush and J. Byers, "Robust identification of shared losses using end-to-end unicast probes," in *Proc. IEEE Int. Conf. Network Protocols*, Osaka, Japan, Nov. 2000, *Errata* available as Boston University CS Technical Report 2001-001.

[8] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," in *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000.

[9] S. Ratnasamy and S. McCanne, "Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements," in *Proceedings of IEEE INFOCOM 1999*, New York, NY, March 1999.

[10] D. Rubenstein, J. Kurose, and D. Towsley, "Detecting shared congestion of flows via end-to-end measurement," in *Proc. ACM SIGMETRICS 2000*, Santa Clara, CA, June 2000.

[11] J. Kurose and K. Ross, *Computer Networking*, Addison Wesley, 2001.

[12] *Netdyn*, http://www.cs.umd.edu/projects/netcalliper/NetDyn.html.

[13] M.F. Shih and A.O. Hero, "Unicast inference of network link delay distributions from edge measurements," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, May 2001.

[14] P.J. Green, "Penalized likelihood," *Encyclopedia of Statistical Sciences, update volume*, vol. 3, pp. 578–586, 1999.

[15] E. Kolaczyk and R. Nowak, "A multiresolution analysis for likelihoods: Theory and methods," submitted to Annals of Statistics, 2000.

[16] N. Duffield and F. Lo Presti, "Multicast inference of packet delay variance at interior network links," in *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000.

[17] V. Jacobson, "pathchar," 1997, ftp://ftp.ee.lbl.gov/pathchar/msri-talk.ps.gz.

[18] M.F. Shih and A.O. Hero, "Unicast inference of network link delay distributions from edge measurements," Tech. Rep., Comm. and Sig. Proc. Lab. (CSPL), Dept. EECS, University of Michigan, Ann Arbor, May 2001.

[19] V. Jacobson, "traceroute," 1989, available as ftp://ftp.ee.lbl.gov/traceroute.tar.Z.

[20] M. Coates, M. Gadhiok, R. King, and R. Nowak, "`nettomo`: A tool for unicast network tomography," Tech. Rep. TREE-05, Rice University, Jun. 2001.

[21] S. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proc. IEEE Infocom*, 1999.

[22] V. Paxson, "On calibrating measurements of packet transit times," in *Proc. ACM SIGMETRICS 1998*, Madison, 1998.

[23] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Wiley, New York, 1997.

[24] D.W. Scott, *Multivariate Density Estimation: Theory, Practice and Visualization*, Wiley, New York, 1992.

[25] B. Frey, *Graphical Models for Machine Learning and Digital Communication*, MIT Press, Cambridge, 1998.

[26] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artificial Intelligence*, vol. 29, pp. 245–257, 1986.

[27] D.J. Spiegelhalter, "Probabilistic reasoning in predictive expert systems," in *Uncertainty in Artificial Intelligence*, L.N. Kanal and J.F. Lemmer, Eds., pp. 47–68. North Holland, Amsterdam, 1986.